International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-3 Issue-5, June 2014

Privacy-Preserving Location Query Service using Privacy Preserving Distance Computation

Avinash Taskar, Bhushan Sinkar

Abstract—Today's world is none of other than Smart Phones, Tablets, and High Speed Media, and its context-rich functionalities attract considerable users. Many LBS providers use users' location information to offer them convenience and useful functions. However, the LBS could greatly breach personal privacy because location itself contains much information. Hence, preserving location privacy while achieving utility from it is still an challenging question now. This paper tackles this non-trivial challenge by designing a suite of novel fine-grained Privacy-preserving Location Query Protocol (PLQP). Our protocol allows different levels of location query on encrypted location information for different users, and it is efficient enough to be applied in mo bile platforms.

Keywords:- LBS, protocol, PLQP.

I. INTRODUCTION

Location Based Service (LBS) has become one of the most popular mobile applications due to the wide use of smartphones. The smartphones, equipped with GPS modules, have powerful computation ability to process holders' location information, and this brought the flood of LBS applications i n the smartphone ecosystem. A good example is the smartphone camera: if one takes a photo with a smartphone camera, the location where the photo is taken is embedded in the picture automatically, which helps one's remembrance. Furthermore, the explosive growth of social network services (SNS) also assisted its growth by constructing connections between location information and social network. When a picture taken by a smartphone (location embedded) is uploaded to the Facebook album, the system automatically shows the location of the picture on the map, and this is shared with the owner's friends in the Facebook (unless the privacy setting specifie s otherwise). Many similar applications exploit both LBS and SNS. They offer several attractive functions, but location information contains much more information than barely the location itself, which could lead to unwanted information leakage. For example, when Alice and Bob both use check-in application in Facebook (which leaves a location record in one's webpage) in a nice restaurant, it is inferable that they are having a date and that they could be in a relationship. This inference might be an unintended information leakage from Alice's and Bob's perspective. Therefore, a privacy-preserving protocol is needed to prevent significant privacy breach resulted from t he combination of LBS and SNS.

Manuscript published on 30 June 2014.
***** Correspondence Author (s)
Avinash Taskar, Department of Computer Science, Sandip Institute of Engineering & Management, Mahiravani, Nashik, Maharashtra, India.
Mr. **Bhushan Sinker**, Sparhasoft Technologies, Nashik, Maharashtra, India.

The simplest way, which most of applications adopted, is to exert group based access control on published locations: specify a group of user who can or cannot see them. Social photo sharing website Flickr only let users choose all users, neighbors', friends or family to allow the access to the locations, and SNS websites Facebook and Google+ additionally support custom groups to specify the accessible user groups. Mobile applications are much worse. Many mobile applications (e.g., Circle, Who's around and Foursquare) even do not offer group choices to the users, instead, they only ask users whether they want to disclose the location or not. Obviously, this is too simple to achieve what users need. First of all, from users' perspective, it is hard to explicitly determine a user group such that their locations are visible only to them. It is more natural to find a condition such that friends who satisfy it can or cannot see the location. Secondly, binary access control (can or cannot) is far beyond enough to properly configure the privacy setting. In the previous example of the two lovers Alice and Bob, Alice might want to share her date at the restaurant with her best friends and discloses the exact location to them. Besides, Alice might also want other friends to know that she is having a good time in downtown, but not detailed location. In this case, approximate settings between `can' and `cannot' are needed to fulfil her requirements. As discussed above, existing privacy control settings in LBS are `coarse' in the sense that: 1) users can only explicitly specify a group of users who can or cannot access the location information; 2) access control policy supports binary choices only, which means users can only choose to enable or disable the information disclosure. The existing control strategies also suffer from privacy leakage in terms of the server storage. Even if one disables all of the location disclosure, his location is still open to the server, which in fact is users' top con-cern. Therefore, a fine-grained privacy control executable on encrypted location data is needed to further foster the LBS and its related business market.

A. Contributions

This paper proposes a fine-grained Privacy-preserving Location Query Protocol (PLQP) which enables queries to get location information (e.g., Searching a friend's approximate location, Finding nearest friends) without violating users location privacy. This is not a trivial job since simple anonymization makes it impossible to utilize them for queries.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Also, if one directly applies queries or functions on the raw location information, privacy leakage is inevitable. Main contributions of our work are three-fold.

- ***Fine-Grained Access Control*:** Our protocol allows users to specify a condition instead of a group and exert access control over the users who satisfy this condition. This is more scalable since users can simply add a new condition for new privacy setting instead of hand-picking hundreds of users to form a new group. Also, this is more user-friendly because users themselves do not clearly know which of their friends should or should not access the information most of time.
- ***Multi-leveled Access Control*:** The protocol also supports semi-functional encryption. That is, the protocol enables users to control to what extent (or level) others can learn his location. The lowest level corresponds to nothing, and the highest level corresponds to one's exact location. Levels between them correspond to indirect information about one's location.
- ***Privacy-Preserving Protocol*:** In our protocol, every location information is encrypted and queries are processed upon ciphertexts. Therefore, a location publisher's friends learn nothing but the result of the location query, which is under the location publisher's control. In addition, since every location is encrypted, even the server who stores location information does not learn anything from the ciphertext.

II. RELATED WORK

There are several works achieving privacy-preserving location query [1]–[4], which are based on k-anonymity model. The k-anonymity model [5] has been widely used to protect data privacy. The basic idea is to remove some features such that each item is not distinguishable among other k items. However, relevant techniques which achieve k-anonymity of data cannot be used in our case for the following four reasons: 1) Those techniques protect the privacy of the data stored in servers. In our PLQP, we do not store the data at all. 2) In LBS, location data is frequently updated, and this dynamic behaviour introduces huge overhead to keep the data k-anonymous. 3) As analyzed in Zang *et al.* [6], achieving k-anonymity in location dataset significantly violate the uti lity of it even for small k, so it is not suitable for our location query protocol. 4) k is generally a system-wide parameter which determines the privacy level of all data in the system, but our goal is to leave the decision of privacy level to each user. Kido *et al.* [7] proposed a scheme which appends multiple false locations to a true one. The LBS responds to all the reports, and the client only collects the response corresponding to the true location. They examined this dummy-based tech-nique and predicted how to make plausible dummy locations and how to reduce the extra communication cost. However, their technique protects the users' location privacy against LBS provider. We are also interested in a user's location privacy against other users. In the mix zone model proposed by Beresford *et al.* [8], users are assigned different pseudonyms every time he enters the mix zone, and users' paths are hidden by doing so. Several works [9]–[11] are based on this model, but they guarantee th e privacy only when the user density is high and user behaviour pattern is unpredictable. Also, most of them require trusted servers. There are also works related to CR (cloaking region) [12]–[15]. In these works, the LBS receives a cloaking region instead of actual users' locations. Gedit *et al.* proposed spatial cloaking and temporal cloaking in [12]. Each query specifies a temporal interval, and queries within the same interval, whose sources are in the vicinity of the first query's source, are merged to a single query. Otherwise, the query is rejected because it has no anonymity. Kalnis *et al.* [13] used the Hilbert space filling curve to map the two dimensional locations to one dimensional values, which are then indexed by a B+ tree. Then, they partition the one dimensional sorted list into groups of n users, which is the CR of their scheme. Since this Hilbert Cloaking is not based on geometric space, it guarantees privacy for any location distribution. However, a certain range, where the user is located, is disclosed in CR-based approaches, and this is out of users' control. It is more desirable to allow users themselves to configure it.

III. SYSTEM MODEL AND PROBLEM FORMULATION

A. System Model

We denote every person engaged in the protocol as a user U_i (we do not differentiate smartphone users and PC users), the user who publishes his location as a publisher P_i and the user who queries the location information of other user as a querier Q_i. Note that a user can be a querier and a publisher at the same time. When he queries on others, he acts as a querier and when he is queried, he acts as a publisher. That is, $U_i = P_i = Q_i$ for the same i. Also, mobile applications or SNS applications which sup-port LBS are denoted as service providers SP. Q and P retrieves keys from SP, which are used for access control. For simplicity, we consider only one SP here. We assume an independent semi-honest model for users and service providers. That is, they all behave independently and will try to extract useful information from the ciphertexts, but they will follow the protocol in general and will not collude with each other. We further assume that every user communicate with each other via an anonymized network (e.g., Tor: https://www.torproject.org) or other anonymized protocol ([16]) such that the privacy is not compromised by the underlying network protocol. We assume the origin of a packet is successfully hidden, which is out of this paper's scope (otherwise any attacker can achieve the location based on the origin of the packet).

B. Location Assumption

For simplicity, we assume the ground surface is a plane, and every user's location is mapped to an Euclidean space with integer coordinates (with meter as unit). That is, ev-eryone's location can be expressed as a tuple of coordinates representing a point in a grid partition of the space.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

This does not affect the generality since there exists a bijection between spherical locations and Euclidean locations.By approximating the coordinates in the Euclidean space to the nearest grid point, we can show that it results in errors of the Euclidean distance $\sqrt{\ }$ between two locations at most 2 meters when the space is partitioned using grid of side-length 1 meter.

The Euclidean distance between two users with locations $\mathbf{x}_1 = (x_{11}, x_{12}, x_{13})$ and $\mathbf{x}_2 = (x_{21}, x_{22}, x_{23})$ is $\mathbf{s}$ $\text{dist}(U_1, U_2) = |\mathbf{x}_1 - \mathbf{x}_2| = \sqrt{\sum_{i=1}^{3} (x_{1i} - x_{2i})^2}$. Given a real location on the surface of the earth, we need to compute the surface distance, denoted as SD(Ui, Uj), between these two points. By assuming that the earth is a sphere with radius meters, it is easy to show that $\text{SD}(U_i, U_j) = R\,\text{dist}(U_i, U_i)$ $2 \arcsin(\frac{2R}{\ }) \cdot R$. Then the surface distance can be quickly computed from the Euclidean distance. To check if the surface distance satisfies certain conditions, we can co nvert it to check if the Euclidean distance satisfying corresponding conditions. For example, dist(U1, U2) ≤ D is equivalent as SD(Ui, Uj) ≤ 2R arcsin(D/2R). For simplicity and convenience of presentation, in this paper, we will focus on the Euclidean distance instead of the surface distance. Notice that although we consider only Euclidean space here, our protocol works for any system where distance is a polynomial of location points x's, where x is a vector.

C. Problem Statement

Each user Ui has his location information xi = (xi1, xi2 , xi3) which determines his current location. He also has an attribute set Si which determines his identity (e.g., Uni-versity:I.I.T, Degree:Ph.D, Major:Computer Science). Then, a querier Qi uses his current location information and attribute set to execute a query (function) f on a publisher Pj 's location information xj . According to Qi's location information xi and his attribute set Si, he obtains the corresponding query result f (xi, Si, xj). Note that different xi and Si leads to different level of query result. During the whole protocol, Qi or Pj cannot learn any useful extra information about each other's location information. In this paper, we propose novel protocols such that the location publisher exerts a fine-grained access control on w ho can access what location information. For example, a publisher could specify the following access control policies: (1) a user can know which city I am in if s/he is in my friend list; or (2) a user can check whether the distance between him and me is less than 100 meters if s/he is my classmate; or (3) a user can compute the exact distance between us if we both went to the same university. We generally assume that a user Ui has a set of attributes Ai, and that an access control policy of the publisher is specified by a boolean function (specified as an access tree T) on all possible attributes of users. According to the location information disclosed to the querier, we define four different levels of queries.

Definition 1. Level 1 Query*: When the query ends,* Q *learns whether* dist(Q, P) ≤ τ *or not if the attributes of the querier satisfy a certain condition specified by the publisher, wher e* τ *is a threshold value determined by* P. *The querier knows nothing else about the location of the publisher.*

Definition 2. Level 2 Query*: When the query ends,* Q *learns whether* dist(Q, P) ≤ τ *when the attributes of the querier satisfy a certain condition specified by the publisher, wher e* τ *is a threshold value determined by* Q. *The querier knows nothing else about the location of the publisher.*

Definition 3. Level 3 Query*: When the query ends,* Q *learns the* dist(Q, P) *if the attributes of the querier satisfy a certain condition specified by the publisher. The querier knows noth ing else about the location of the publisher.*

Definition 4. Level 4 Query*: When the query ends,* Q *learns the function* F (**x**) *of the location* ***x*** *of* P *if the attributes of the querier satisfy a certain condition specified by the publish er. Here function* F *is defined by the publisher. The querier knows nothing else about the location of the publisher.*

It is easy to show that the level i query provides better privacy protection than level i + 1 query, for i = 1, 2. Level 4 query provides most information in general. In level 4 query, the function F could be used by the publisher to exert fine-grained access control on his location information. Fo r example F (**x**) could return the city of the location, the zip-code of the location or the exact location information.

IV. BACKGROUND

In our Privacy-preserving Location Query Protocol (PLQP), various cryptographic concepts are used. We introduce each of them in this section.

A. Attribute-Based Encryption (ABE)

As Jung et al. discussed in detail in their work [17], in the Attribute-Based Encryption (ABE) [18], the identity of a person is viewed as a set of attributes. This enables the encrypter to specify a boolean function to do access control. There are two types of ABE system: Goyal et al.'s Key-Policy Attribute-Based Encryption [19] and Bethencourt et al.'s Ciphertext-Policy Attribute-Based Encryption [20]. The KP-ABE specifies the encryption policy in the decryption key , and the CP-ABE specifies the policy in the ciphertext. Due to many reasons discussed in [17], we will employ CP-ABE as a component of access control.

***1) Access Tree* T** *:* In most of previous ABE works (e.g., [19] [20] [21]), encryption policy is described with an access tree. Each non-leaf node of the tree is a threshold gate by a threshold value θ, and each leaf node x is described by an attribute. A leaf node is satisfied if a key contains the corresponding attribute, and a non-leaf threshold gate is satisfied if at least θ children are satisfied. Note that this threshold-gate based access tree is able to express arbitrary condition, which makes the privacy control in our protocol flexible and scalable.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

2) Definition: With the access tree defined as above, the CP-ABE scheme is defined as follows:

Setup → PK, MK. The setup algorithm takes nothing as input other than the implicit security parameter. It outputs the public parameter PK and a master key MK. The master key belongs to the key issuer and is kept secret.

Encrypt(PK, M , T) → E_T (M). The encryption algorithm takes as input the public key PK, a message M , and an access tree T . It will encrypt the message M and returns a ciphertext CT such that only a user with key satisfying the access tree T can decrypt it.

KeyGenerate(PK, MK, S) → SK. The Key Generation algorithm takes as input the public key PK, the master key MK and a set of attributes S. It outputs a private key SK which contains the attributes in S.

Decrypt(PK, SK, E_T (M)) → M . The decryption algo-rithm takes as input the public parameter PK, a private key SK whose attribute set is S, and a ciphertext CT which contains an access tree T . It outputs the original message M if and only if the set S satisfies the access tree T . We direct the readers to [20] for detailed construction.

B. Homomorphic Encryption (HE)

Homomorphic Encryption (HE) allows direct addition and multiplication on ciphertexts while preserving decryptability. That is, following equations are satisfied (Note that this is only an example of a HE, and detailed operations vary for different HE system).

$$\mathrm{Enc}(m_1) \cdot \mathrm{Enc}(m_2) = \mathrm{Enc}(m_1 + m_2)$$
$$\mathrm{Enc}(m_1)^{\mathrm{Enc}(m2)} = \mathrm{Enc}(m_1 \cdot m_2)$$

where Enc(m) stands for the ciphertext of m.

In general, there are two types of HE: Partially Homo-morphic Encryption (PHE) and Fully Homomorphic Encryp-tion (FHE). PHE supports constant number of additions and multiplications, and FHE supports unlimited additions and multiplications but it is much less efficient than PHE. As discussed by Lauter *et al.* in [22], the decryption time of FHE system is too high to be used in a real application, and in most of cases one only needs a few number of multiplications or additions. Therefore, Pallier's system, which is much simpler and thus efficient, is our choice: it involves only one multiplication for each homomorphic addition and one exponentiation for each homomorphic multiplication.

1) Definition of Paillier's Cryptosystem: Paillier's cryp-tosystem is composed of three algorithms – KeyGenerate, Encrypt and Decrypt.

KeyGenerate → EK, DK. An entity randomly chooses two large prime numbers p and q of same bit length. He then computes n = pq and $\lambda = (p-1)(q-1)$. Next, he sets $g = (n+1)$ and $\mu = (\lambda \bmod n^2)^{-1} \bmod n$. Then, the encryption key is EK = (n, g) and the decryption key is DK = (λ, μ).

Encrypt(EK, m) → E(m, r). The encrypter selects a ran-dom integer $r \in Z_n$ and computes the ciphertext

$$E(m, r) = g^m \cdot r^n \bmod n^2$$

and publishes it.

Decrypt(E(m, r), DK) → m. The holder of DK = (λ, μ) can decrypt the ciphertext E(m, r). He computes the following to recover the message:

$$m = L((E(m, r))^{\lambda} \bmod n^2) \cdot \mu \bmod n$$

where $L(a) = (a-1)/n \bmod n$.

The Paillier's cryptosystem satisfies the following homo-morphic properties:

$$E(m_1, r_1) \cdot E(m_2, r_2) = E(m_1 + m_2, r_1 r_2) \bmod n^2$$
$$E(m_1, r_1)^{m2} = E(m_1 \cdot m_2, r_1^{m2}) \bmod n^2$$

Note that DK can decrypt only the ciphertexts encrypted with EK which pairs with it. Also, the random number r in a ciphertext E(m, r) does not contribute to decryption or other homomorphic operation. It only prevents the dictionary attack by randomizing the ciphertext. For sake of simplicity, we use E(m) instead of E(m, r) in the remaining paper.

C. Functional Encryption (FE)

Functional Encryption (FE) is a new encryption scheme recently proposed after the Attribute-Based Encryption (ABE). To the best of our knowledge, the concept is first proposed by Boneh *et al.* in [23]. In the open direction of their work, they proposed the terminology `Functional Encryption' and its general concept, and later in 2011, Boneh *et al.* formally defined it and discussed its challenge [24]. According to the ir study, the FE is defined as follows: FE is an encryption scheme such that a key holder can learn a specific function of the data based on the ciphertext, but nothing else about the data. This is totally different from the traditional encryption scheme in terms of the differentiated decryption. In traditional encryption schemes (e.g., PKI, ABE), decryption result of a ciphertext for every authorized users is same: the plaintext. In FE, encrypter can specify a function for each key such that each decryption result is the corresponding function of the plaintext. There are a few recent works related to FE ([25], [26]). However, they mainly focus on hiding encryption policy from ordinary users. To the best of our knowledge, there is no formal construction of FE which satisfies the definition of FE [24].

V. PRELIMINARY DESIGN

In our PLQP, we require that a publisher could specify several access control structures for all potential location queriers. Different access trees will allow access to different level of knowledge about the location information, which is achieved by using FE in our protocol. However, strictly speaking, the encryption in our protocol is not a formal FE because we only support a constant number of functions of the data, so we refer to it as semi-functional encryption. To allow a set of possible queries by all users, we first present distance computation and comparison algorithms which will be used to provide four levels of functions over location data in our semi-functional PLQP.

1) Privacy Preserving Distance Computation: Let $\mathbf{x} = (x_1, x_2, x_3)$ and $\mathbf{y} = (y_1, y_2, y_3)$ be a publisher P's and a querier Q's 3-dimensional location respectively. We use Algorithm 1 to let Q securely compute dist(P, Q) without knowing P's coordinates or disclosing his own one.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Algorithm 1 Privacy Preserving Distance Computation

1: Q generates a pair of encryption and decryption keys of Paillier's cryptosystem: EK = (n, g), DK = (λ, μ). We assume n is of 1024-bit length. E_Q denotes the encryption done by Q using his encryption keys.

2: Q generates the following ciphertexts and sends them to P at **x**.

$$E_Q(1),\ E_Q\Big(\sum_{i=1}^{3} y_i^2\Big),\ \{E_Q(y_i) \mid i = 1, 2, 3\},$$

3: P, after receiving the ciphertexts, executes the following homomorphic operations:

$$\{E_Q(y_i)^{-2x_i}\}_{3} = \{E_{Q_2}(-2x_iy_i)\}, \text{ for } i = 1, 2, 3$$

P

4: P computes and sends the following to the querier Q:

$$E_Q\Big(\sum_{i=1}^{3} x_i^2\Big) \cdot E_Q\Big(\sum_{i=1}^{3} y_i^2\Big) \cdot \prod_{i=1}^{3} (E_Q(-2x_iy_i))$$

$$= E_Q\Big(\sum_{i=1}^{3} (x_i - y_i)^2\Big) = E_Q(|\mathbf{x} - \mathbf{y}|^2)$$

5: Q uses the private key DK to decrypt the $E_Q(|\mathbf{x} - \mathbf{y}|^2)$ to get the distance.

Note that the location **y** is kept secret to P during the whole protocol, since he does not know the private key; on the other hand, the location **x** is also kept secret since Q only achieves $E(|\mathbf{x} - \mathbf{y}|^2)$. However, the location **x** is inferred if Q runs the same protocol at different places for four times in Euclidean space (three times in Euclidean plane). This will be discussed in detail in Theorem VI.1.

2) Privacy Preserving Distance Comparison: Let **x** = (x_1, x_2, x_3) and **y** = (y_1, y_2, y_3) be publisher P's and querier Q's 3-dimensional location respectively. We use Algorithm 2 to let Q learn whether dist(P, Q) is less than, equal to or greater than a threshold value τ , which is determined by the publisher P. The reason δ and δ′ are chosen from Z_2972 and Z_21022 is because otherwise the comparison is not correct due to the modular operations. This will be further discussed in Section VI-F. the extra overhead in the tables. The total overhead should include other regular overhead (control messages, ACKs etc.). In conclusion, the computation and communication overhead of our protocol is low enough to be used in a real mobile network.

VI. CONCLUSION

In this paper, we proposed a fine-grained Privacy-preservin g Location Query Protocol (PLQP), which successfully solves the privacy issues in existing LBS applications and provides various location based queries. The PLQP uses our novel distance computation and comparison protocol to implement semi-functional encryption, which supports multi-levelled access control, and used CP-ABE as subsidiary encryption scheme to make access control be more fine-grained. Also, during the whole protocol, unless intended by the location publisher, the location information is kept secret to anyone else. We also conducted experiment evaluation to show that the performance of our protocol is applicable in a real mobile network.

REFERENCES

[1] T. Hashem and L. Kulik, "Safeguarding location privacy i n wireless ad-hoc networks," *Ubicomp 2007: Ubiquitous Computing*, pp. 372–390, 2007.

[2] C. Bettini, X. Wang, and S. Jajodia, "Protecting privacy against location-based personal identification," *Secure Data Management*, pp. 185–199, 2005.

[3] M. Mokbel, C. Chow, and W. Aref, "The new casper: query processing for location services without compromising privacy," in *Proceedings of the 32nd international conference on Very large data bases*, VLDB Endowment, 2006, pp. 763–774.

[4] K. Vu, R. Zheng, and J. Gao, "Efficient algorithms for k-an onymous location privacy in participatory sensing." in *IEEE INFOCOM*, 2012.

[5] L. Sweeney *et al.*, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty Fuzziness and Knowledge Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[6] H. Zang and J. Bolot, "Anonymization of location data doe s not work: A large-scale measurement study," in *Proceedings of the 17th annual international conference on Mobile computing and networking*, 2011, pp. 145–156.

[7] H. Kido, Y. Yanagisawa, and T. Satoh, "Protection of loca tion privacy using dummies for location-based services," in *21st International Con-ference on Data Engineering Workshops, 2005*, pp. 1248–1248.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)